P&S Heterogeneous Systems

Accelerating Agent-Based Simulations with BioDynaMo

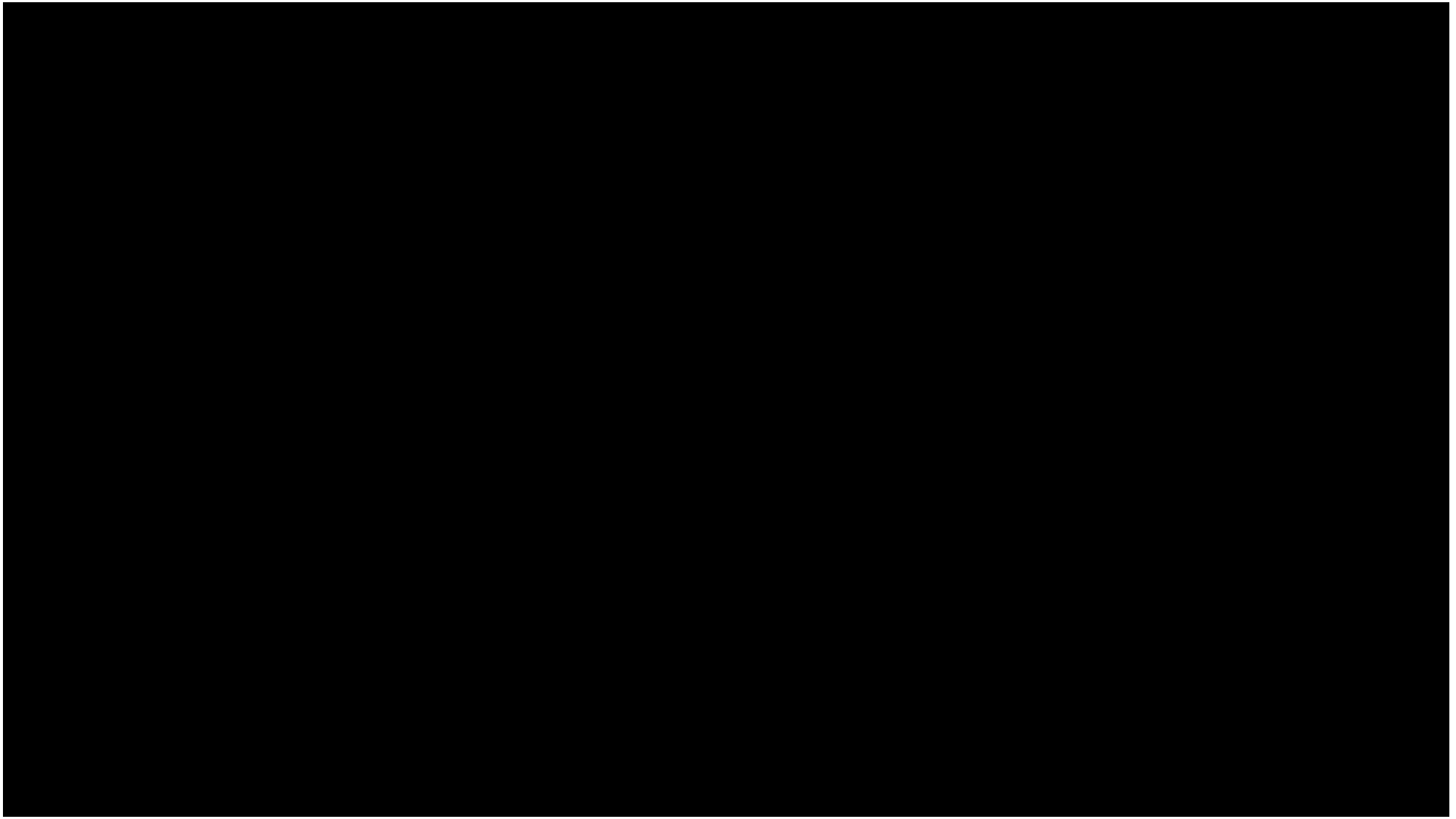
Lukas Breitwieser
ETH Zürich

Fall 2022

30 January 2023

Introduction to Agent-Based Simulation

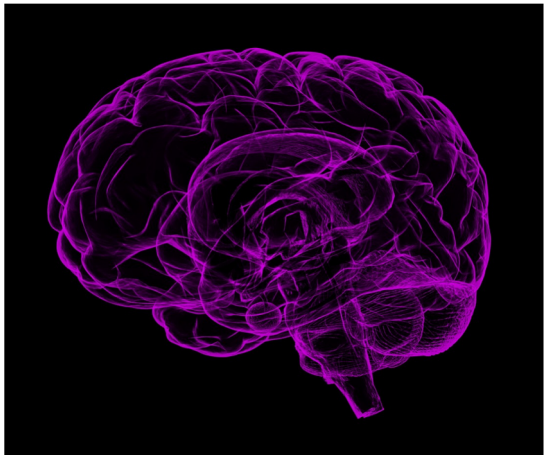
Modeling complex systems

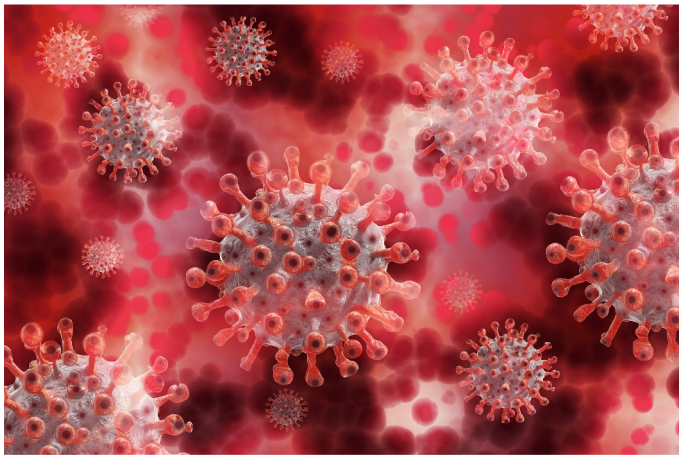


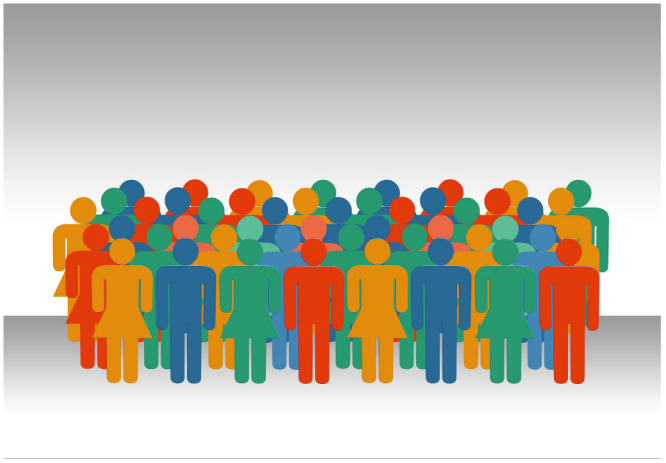
What is agent-based simulation?

- Also called individual-based modeling
- Bottom-up approach
  - Modeling the trees not the forest
- Characteristics
  - Local interaction
  - Emergent behavior

Agent-based simulation is very versatile







Source: Macal and North (2014), https://doi.org/10.1109/WSC.2014.7019874

Agent-based simulation is very versatile (cont.)

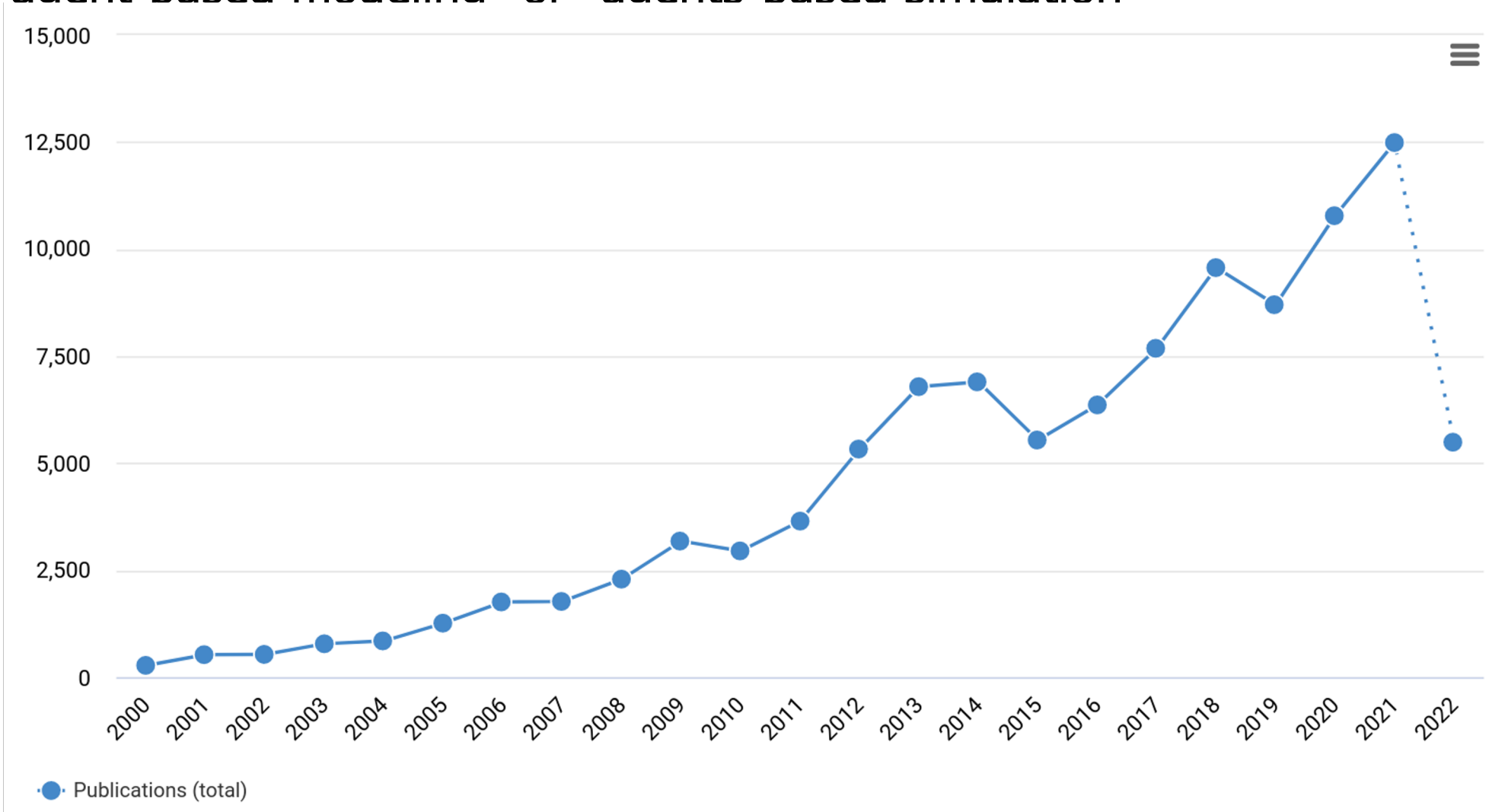
Table 1: A sample of recent agent-based applications

Application Area:	Agent-based Model Focus:
Agriculture	A spatial individual-based model prototype for assessing potential pesticide exposure of farm-workers conducting small-scale agricultural production (Leyk et al. 2009)
Air Traffic Control	Air traffic control to analyze control policies and performance of an air traffic management facility (Conway 2006)
Anthropology	Prehistoric settlement patterns and political consolidation in the Lake Titicaca basin of Peru and Bolivia (Griffin and Stanish 2007)
Biomedical Research	*The Basic Immune Simulator*, to study the interactions between innate and adaptive immunity (Folcik et al. 2007)
Crime Analysis	A realistic virtual urban environment, populated with virtual burglar agents (Malleson 2010)
Ecology	Investigate the trade-off between road avoidance and salt pool spatial memory in the movement behavior of moose (Grosman et al. 2011) Predator-prey relationships between transient killer whales and other marine mammals (Mock and Testa 2007)
Energy Analysis	A building occupant network energy consumption decision-making model (Chen et al. 2011) Application for the Smart Grid (Jackson 2010) Energy investment decision making (Tobias 2008) Oil refinery supply chain (Van Dam et al. 2008)
Epidemiology	Pandemic disease model accounting for individual behavior and demographics (Aleman et al. 2010) Global-scale agent model of disease transmission (Parker and Epstein 2011)
Evacuation	Tsunami evacuation using a modified form of Helbing's social-force model applied to agents (Puckett 2009)
Market Analysis	Consumer marketing model developed in collaboration with a Fortune 50 firm (North et al. 2009) Consumer airline market share (Kuhn et al. 2010) Simulation that models the possibilities for a future market in sub-orbital space tourism (Charania et al. 2006)
Social Networks	Model of email-based social networks, in which individuals establish, maintain and allow atrophy of links through contact-lists and emails (Menges et al. 2008)

Source: Macal and North (2014), https://doi.org/10.1109/WSC.2014.7019874

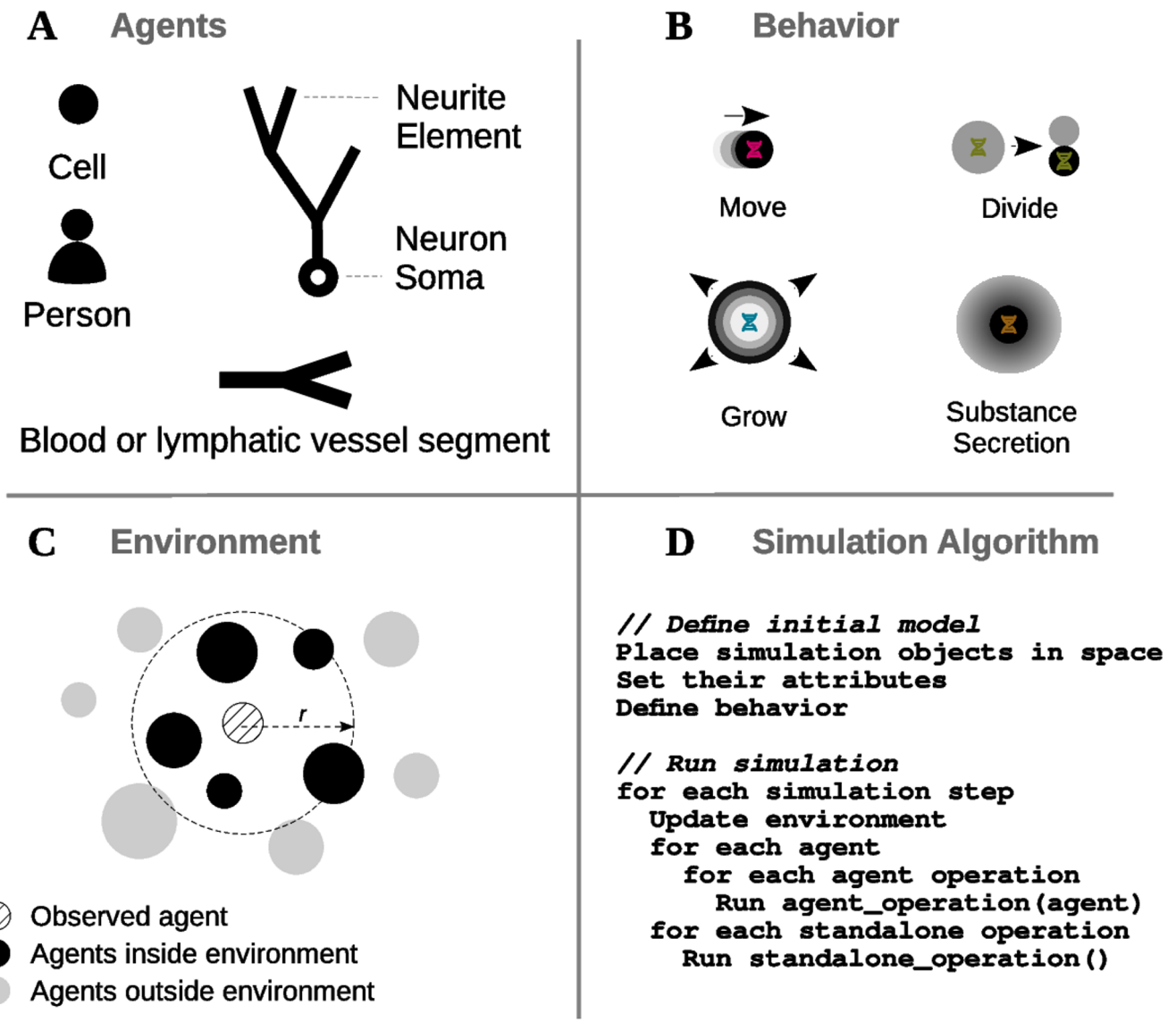
Rising Number of Publications in this field

- Keywords used:
"agent-based modeling" or "agents-based simulation"



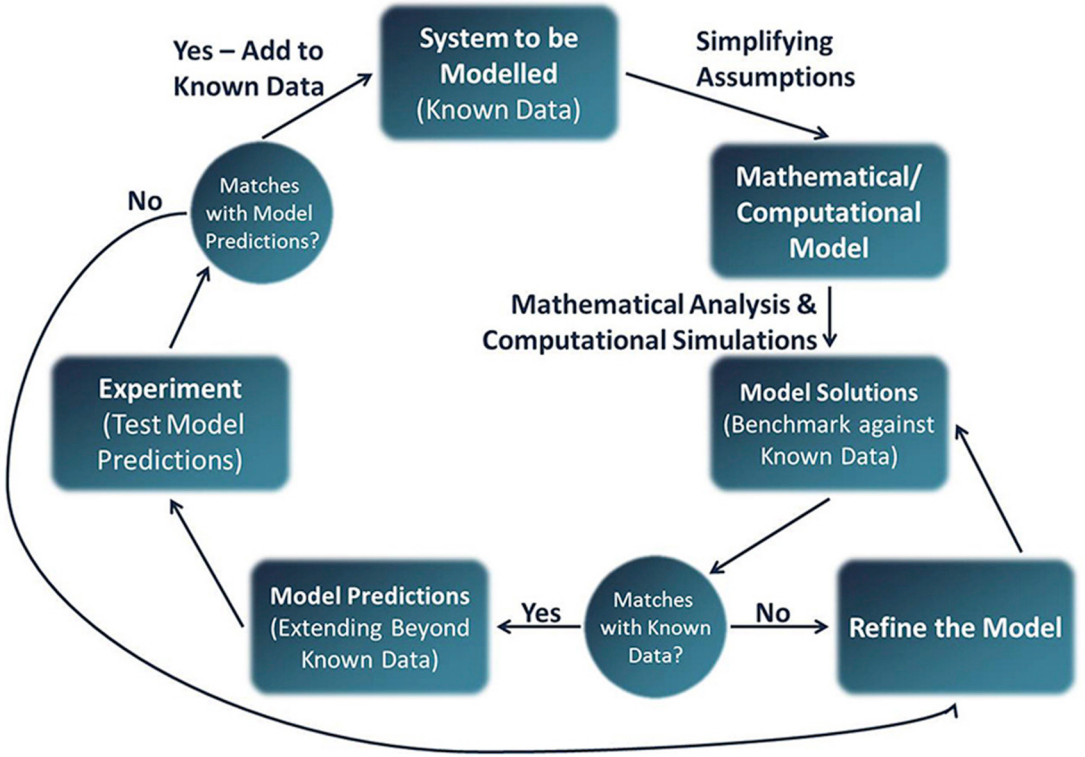
Source: https://app.dimensions.ai

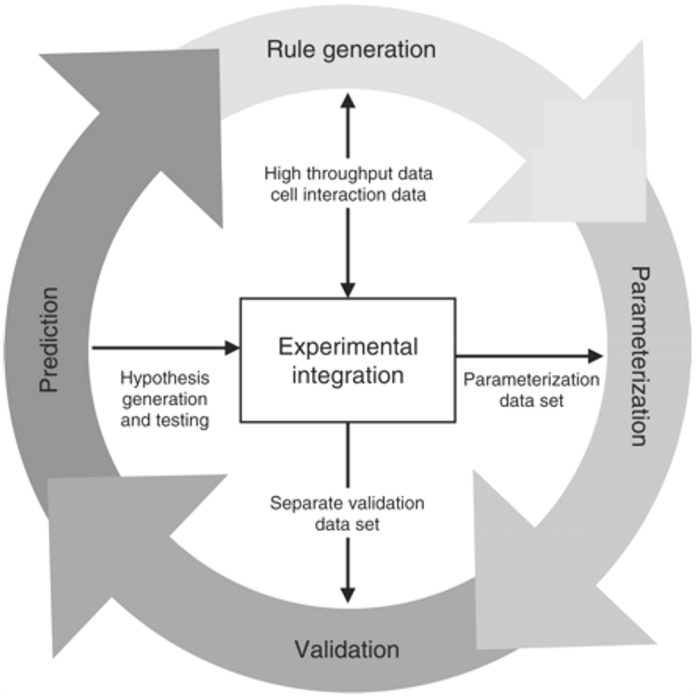
Important building blocks



Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

The process of developing an ABM



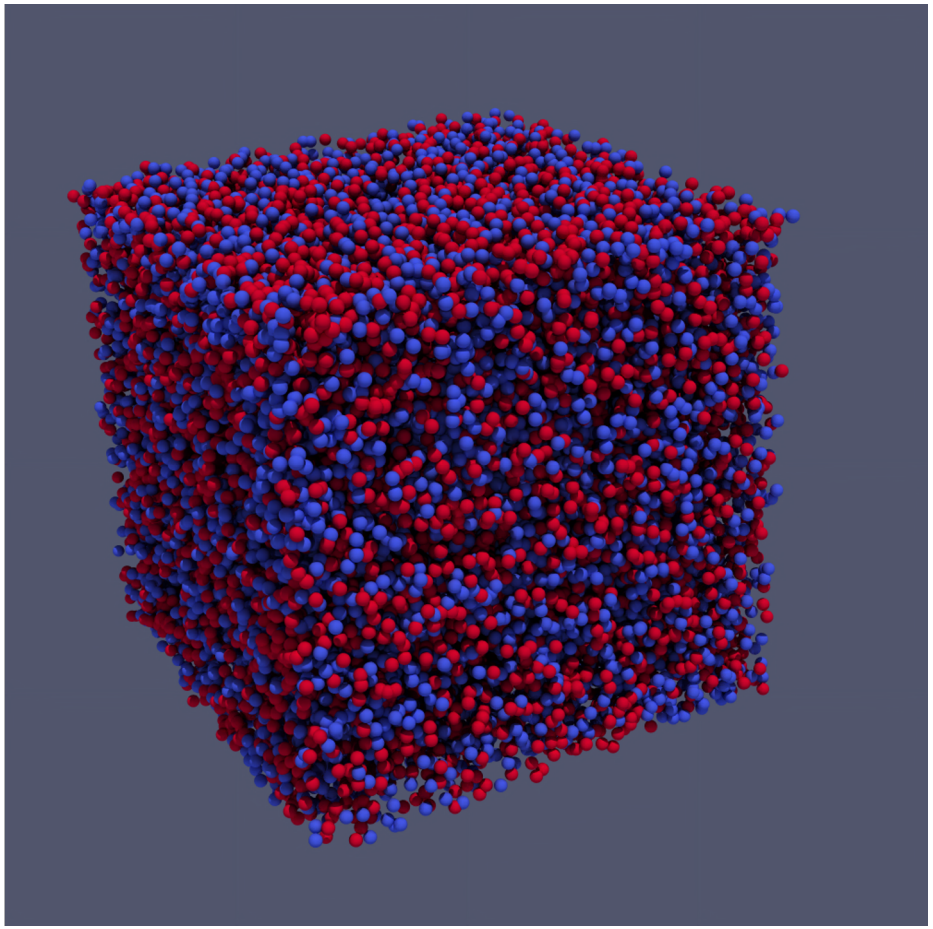


Source: Roberts et al. https://doi.org/10.1016/j.preteyeres.2016.04.001,
Thorne et al. 2007, https://doi.org/10.1093/bib/bbm024

Demo

Cell clustering model

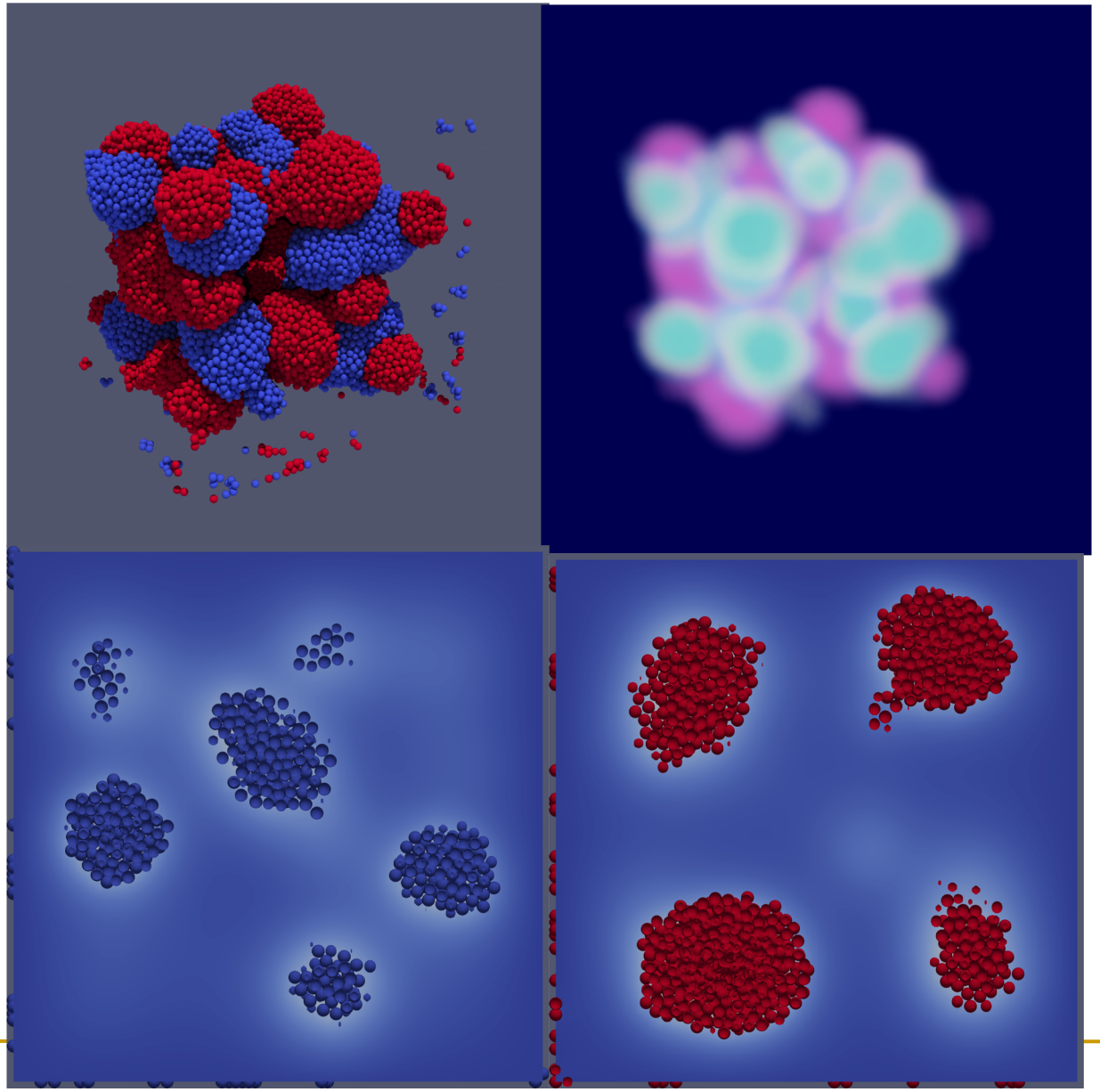
- Agent: Cell
  - Spherical shape
  - cell type
- Behaviors
  - Secrete a substance into the extracellular matrix
  - Follow the concentration gradient (chemotaxis)
- Initial condition
  - Randomly distributed in 3D space



Cell clustering video

https://www.youtube.com/watch?v=jlOk_Y3SUHo

Cell clustering result



Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

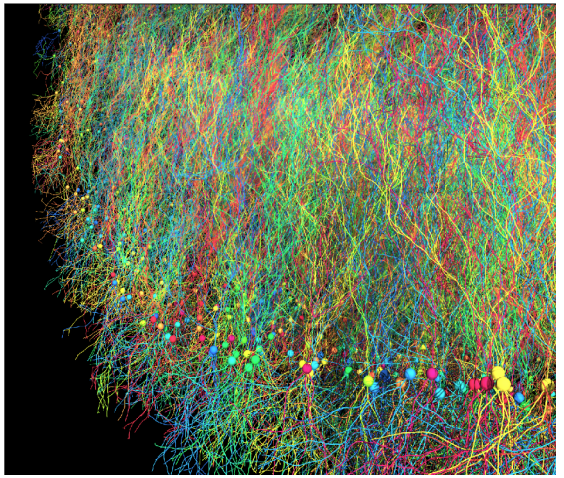
Performance considerations

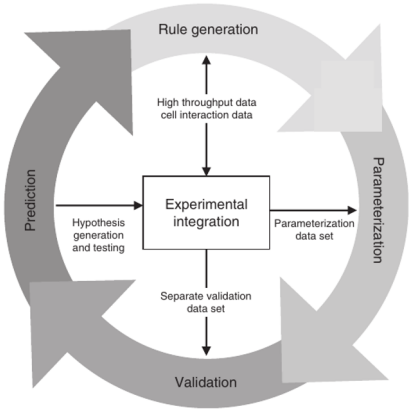
The problem

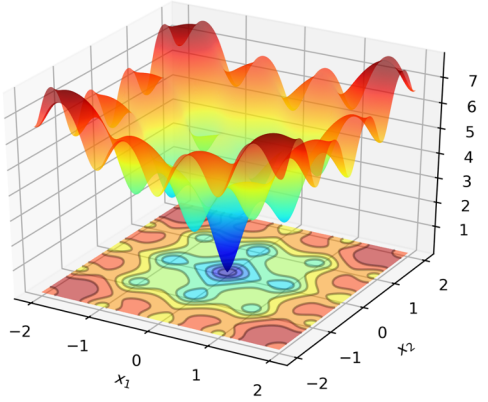
Existing simulation platforms do not always take full advantage of modern hardware.

Impact of low performance

- Limitation of the size and complexity of models
- Longer development time
- Limited capability to explore parameter space
  → less optimal solution
- Increased cost





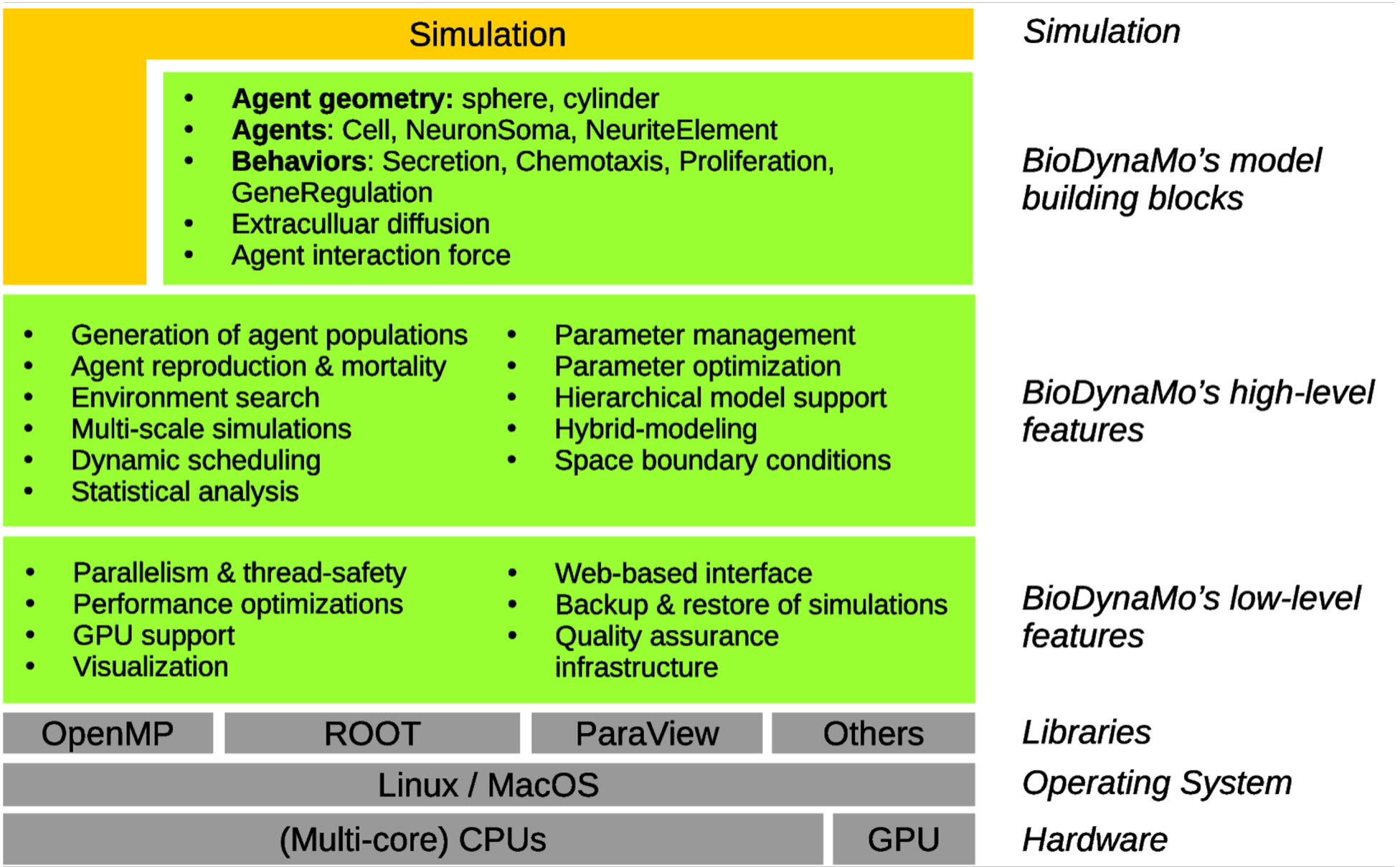


Images credit: https://github.com/kthohr/optim, https://doi.org/10.1093/bib/bbm024

Our solution

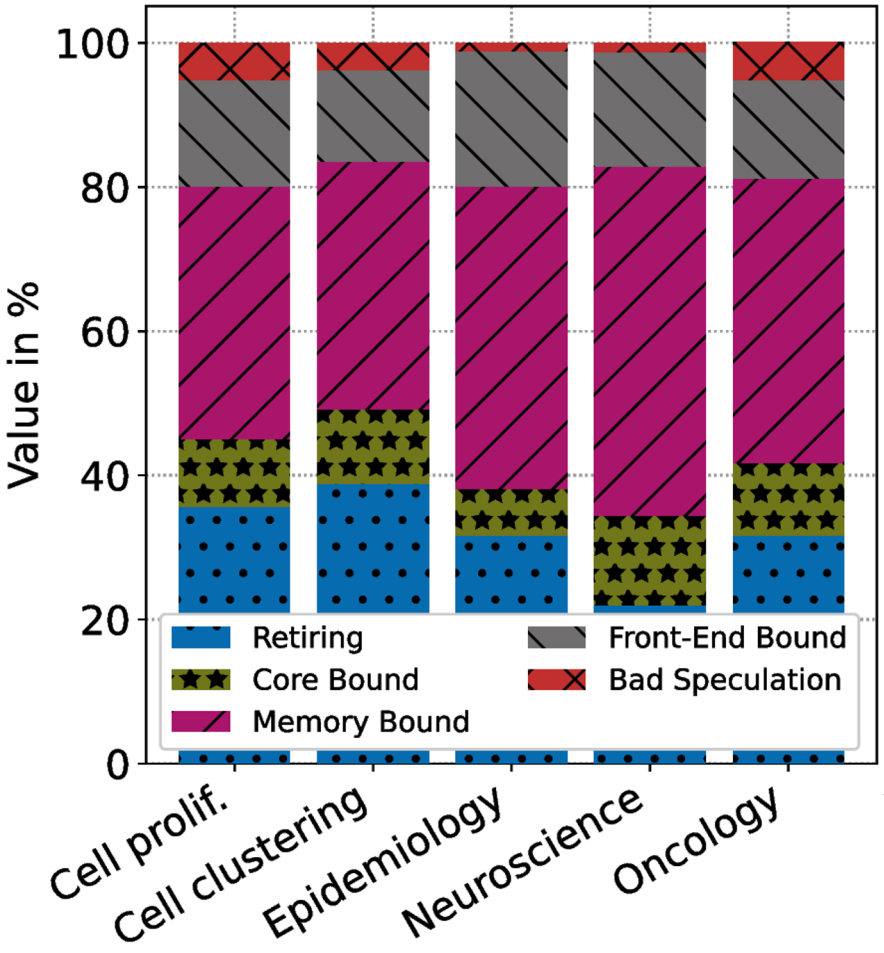
BioDynaMo a **high-performance and modular, agent-based** simulation platform written in C++.

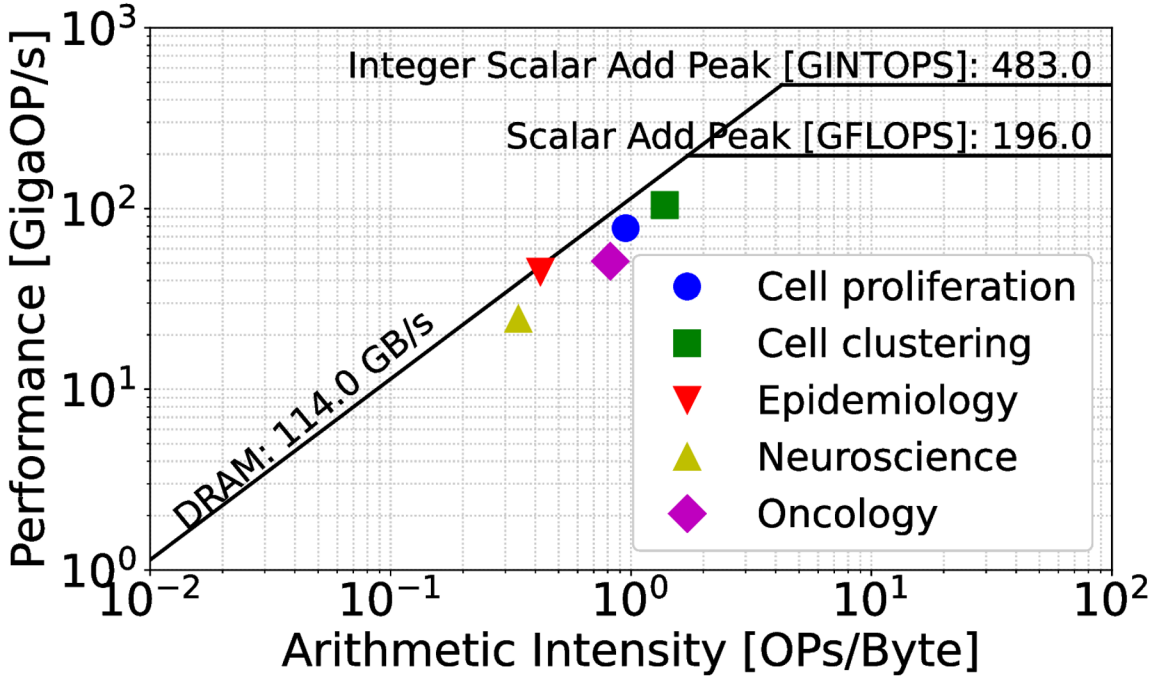
Features and abstraction layers



Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

Challenge: Agent-based workload is memory-bound





Required efficient agent-based primitives

- Iterate over agents and execute operations
- Add and remove agents to/from the simulation
- Efficient environment implementation

Algorithm 1: Agent-based simulation algorithm

```
1  ModelInitialization()
2  for i ∈ iterations do
3  |   for op ∈ pre_standalone_operations do
4  |   |   op();
5  |   end
6  |   wait()
7  |   parallel for a ∈ agents do
8  |   |   for op ∈ agent_operations do
9  |   |   |   op(a);
10 |   |   end
11 |   end
12 |   for op ∈ standalone_operations do
13 |   |   op();
14 |   end
15 |   wait()
16 |   for op ∈ post_standalone_operations do
17 |   |   op();
18 |   end
19 end
```

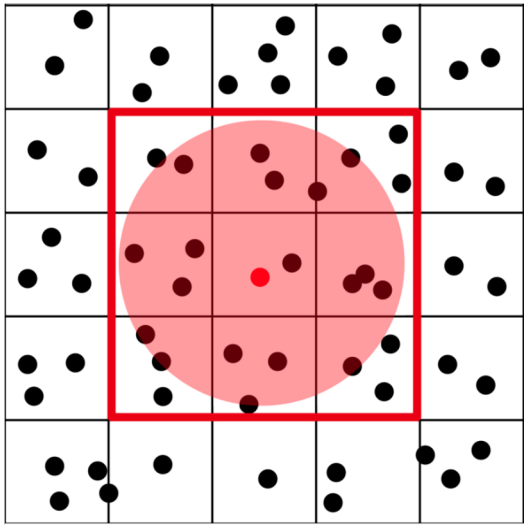
Performance Challenges and Improvements

- Maximize parallelization
  - Optimized uniform grid to search for neighbors
  - Parallelize the addition and removal of agents
- Efficient thread synchronization during agent updates
- Minimize memory access latency
  - NUMA-aware iteration
  - Agent Sorting and Balancing
  - Pool-based memory allocator
- Offload computation to the GPU

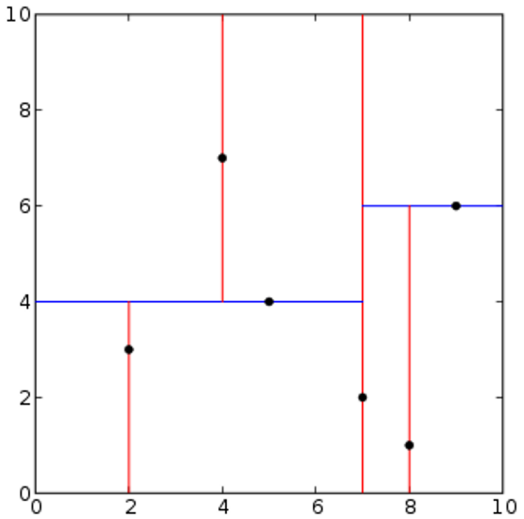
Maximize parallelization

Optimized uniform grid to search for neighbors

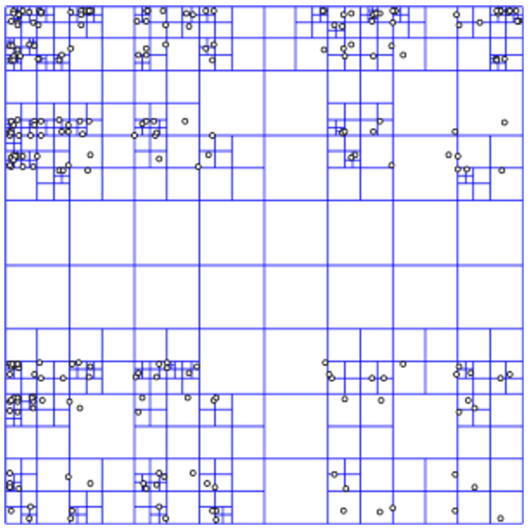
Uniform Grid



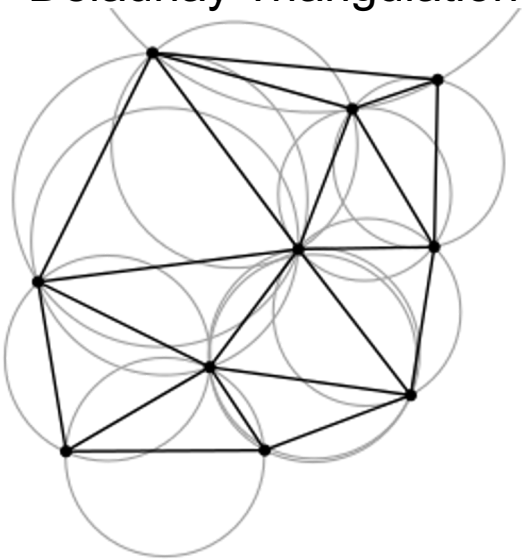
Kd-Tree



Quadtree

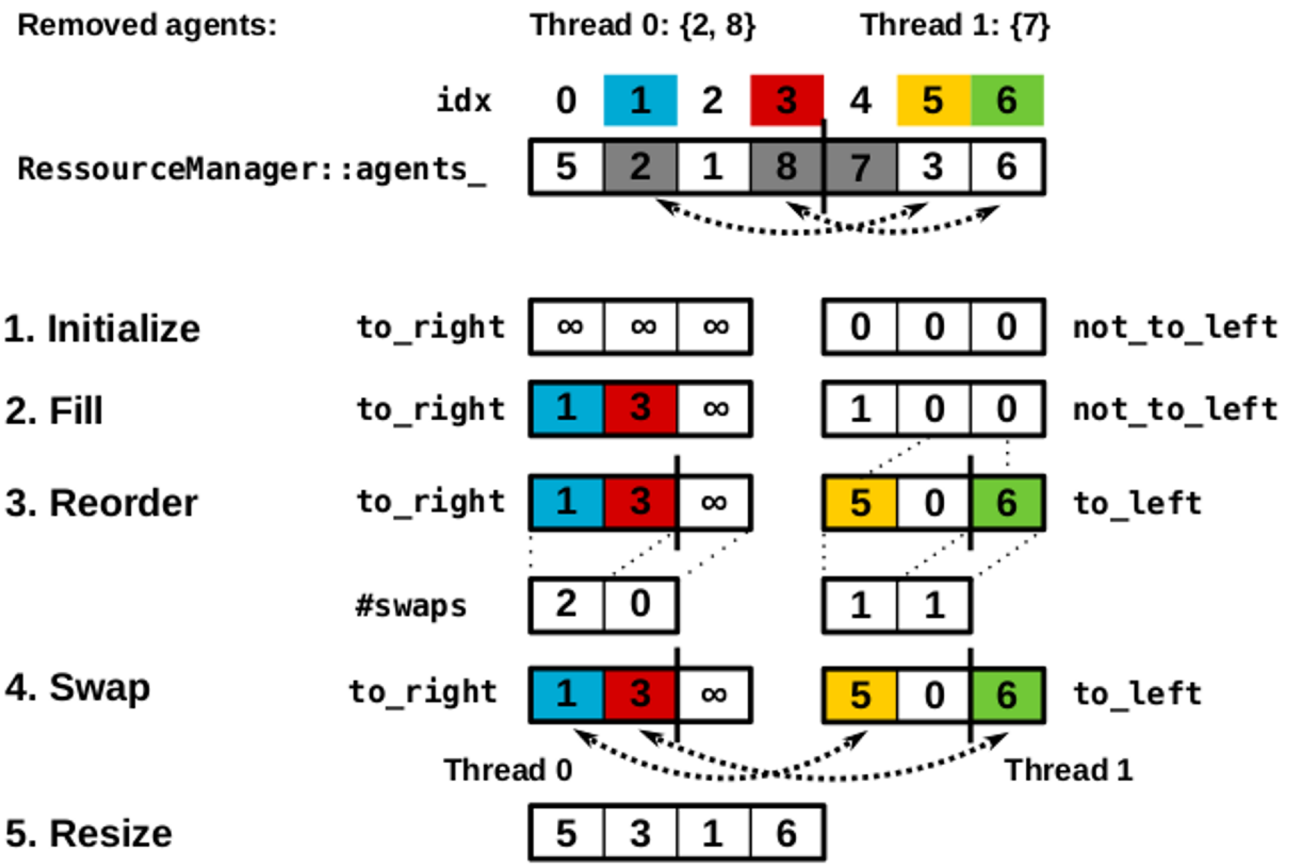


Delaunay Triangulation



Source: Ahmad Hesam

Parallel agent removal mechanism

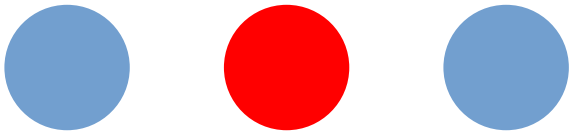


Optimize Thread-Synchronization

Thread-synchronization (TS) during agent-updates

Algorithm 1: Agent-based simulation algorithm

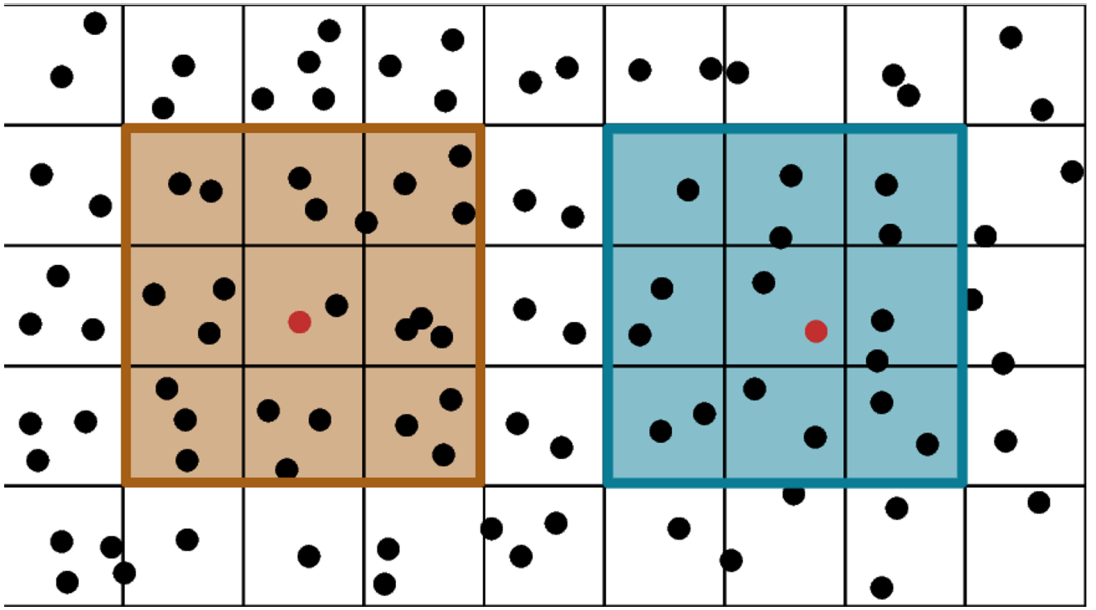
```
1  ModelInitialization()
2  for i ∈ iterations do
3      for op ∈ pre_standalone_operations do
4          op();
5      end
6      wait()
7      parallel for a ∈ agents do
8          for op ∈ agent_operations do
9              op(a);
10         end
11     end
12     for op ∈ standalone_operations do
13         op();
14     end
15     wait()
16     for op ∈ post_standalone_operations do
17         op();
18     end
19 end
```

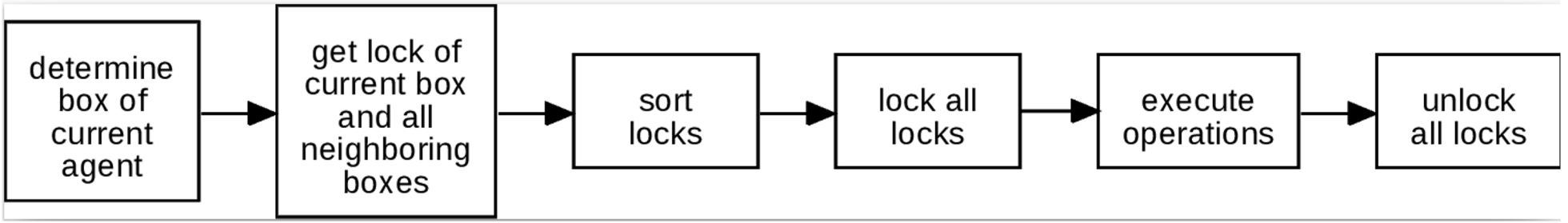


T0 T1

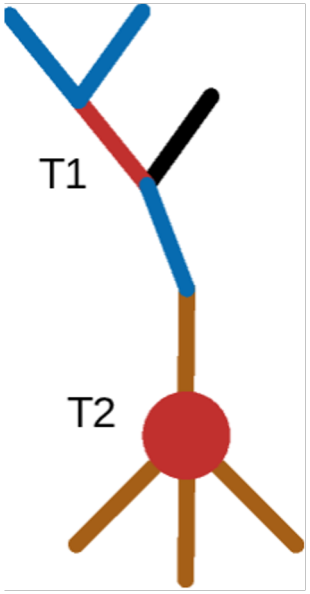
- Only necessary if agents modify their local environment.
  - Two agents (updated by two different threads) could attempt to modify the same neighbor.
- BioDynaMo provides two TS mechanisms
  - Automatic TS
  - User-defined TS

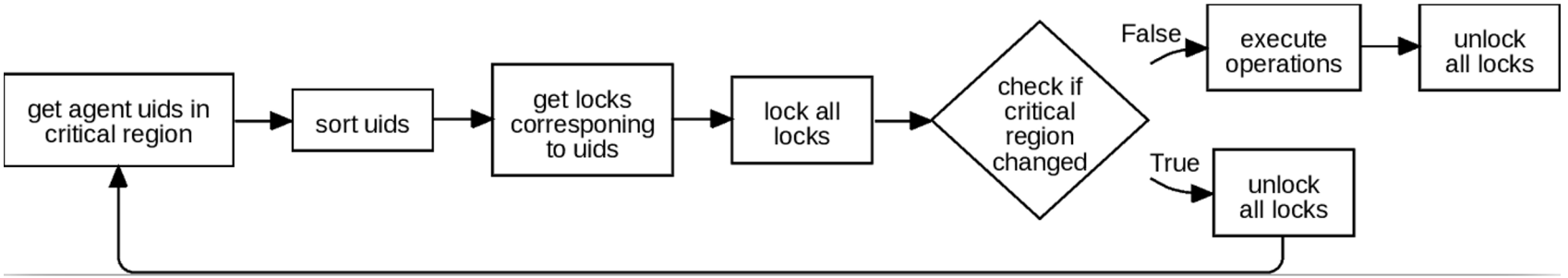
Automatic thread-synchronization





User-defined thread-synchronization

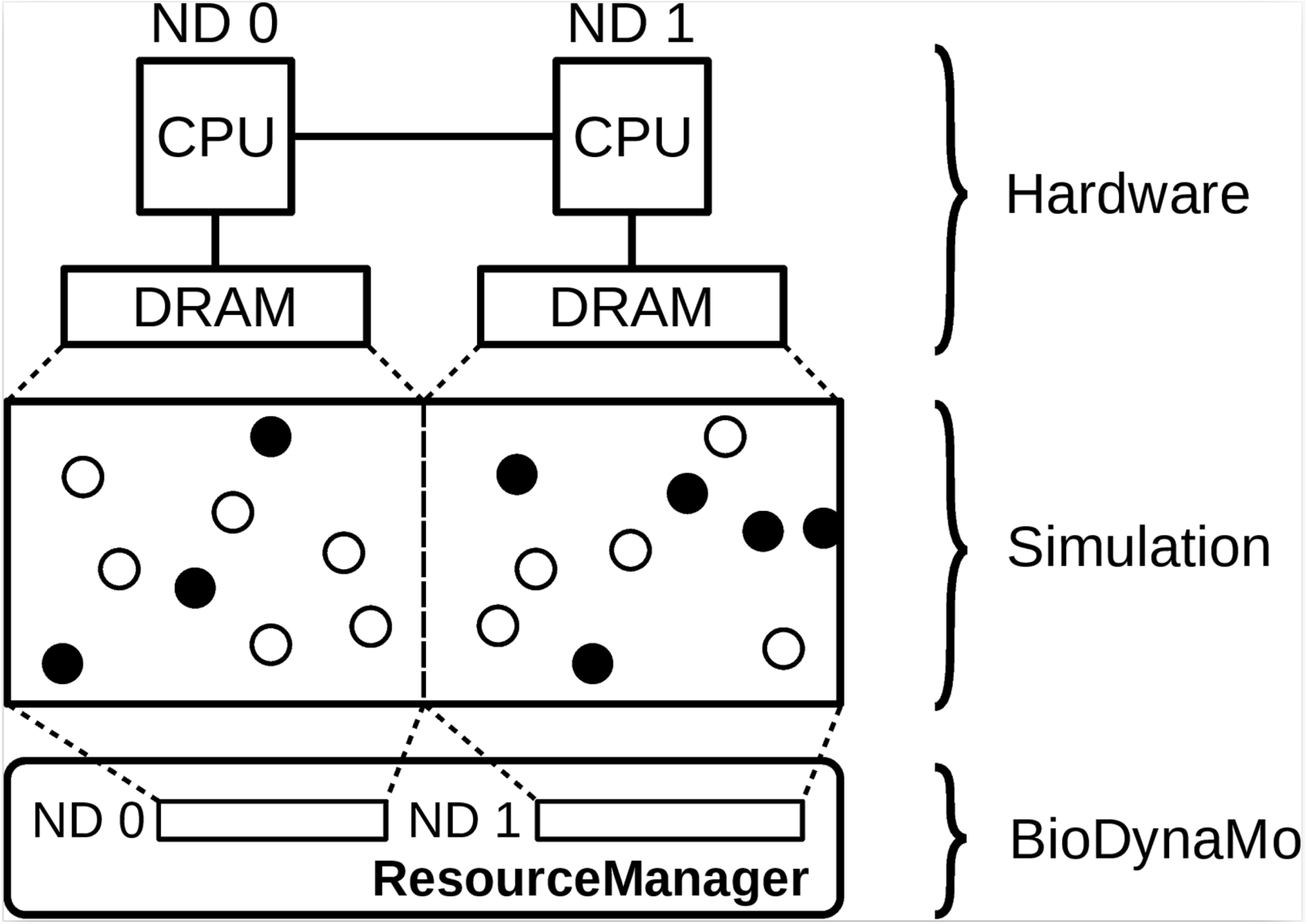




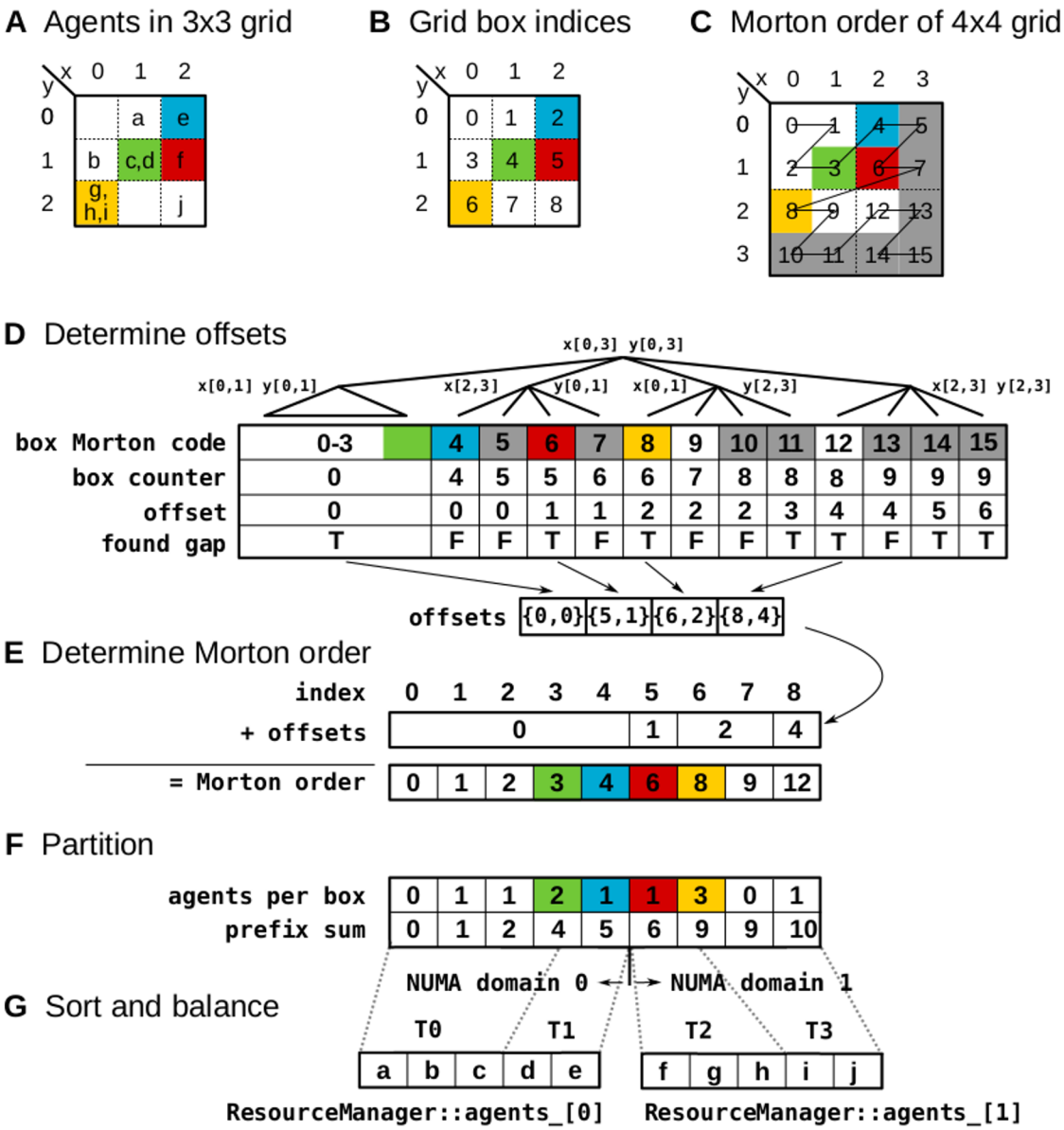

```
void NeuronSoma::CriticalRegion(std::vector<AgentPointer<>>* aptrs) {
  aptrs->reserve(daughters_.size() + 1);
  aptrs->push_back(Agent::GetAgentPtr<>());
  for (auto& daughter : daughters_) {
    aptrs->push_back(daughter);
  }
}
```

Minimize Memory Access Latency

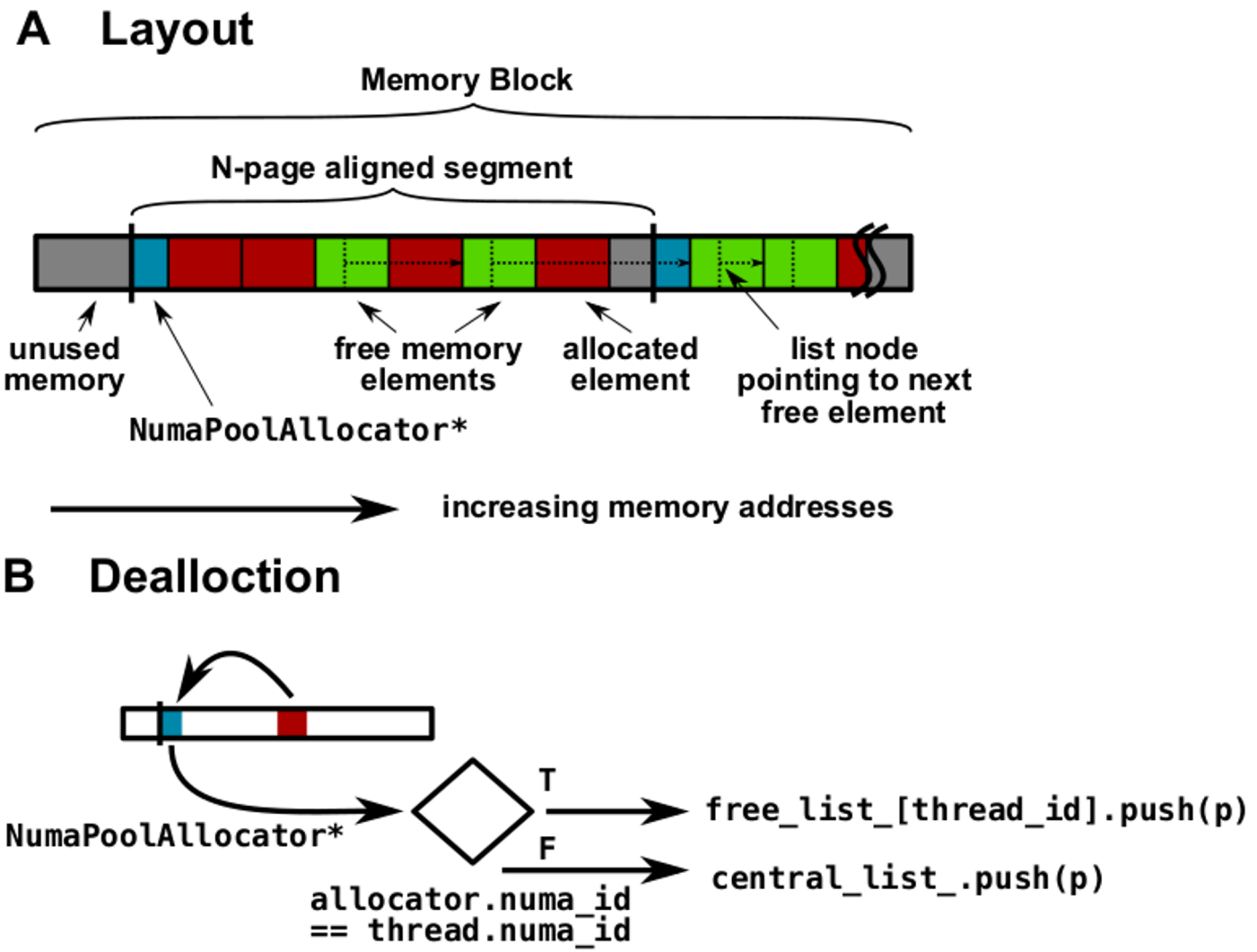
NUMA-aware iteration



Agent sorting and balancing mechanism



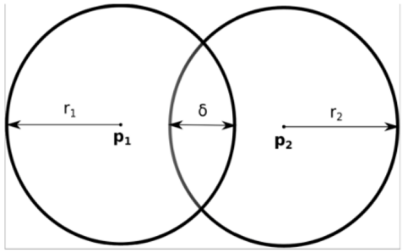
BioDynaMo Memory Allocator



Offload computation to the GPU

BioDynaMo's GPU capabilities

- Operations can have implementations for different compute targets (CPU, GPU, and FPGA).
- If an operation has multiple implementations, the scheduler decides which one to use.
- Currently, BioDynaMo provides a GPU operation to calculate mechanical forces between spheres.



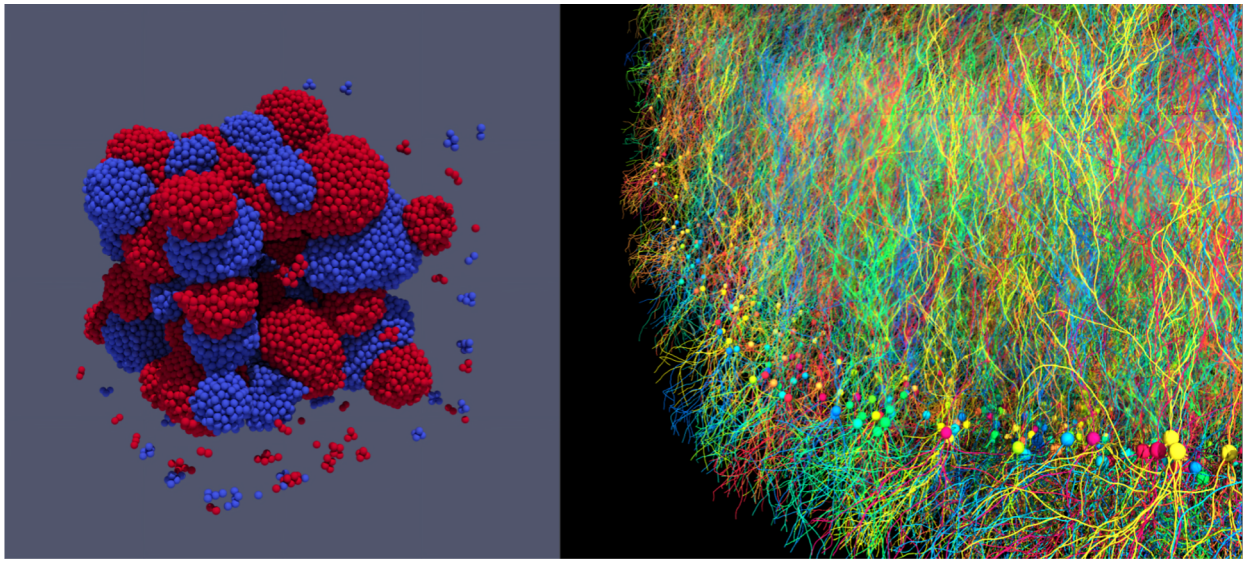
$$\delta = r_1 + r_2 - \|\mathbf{p_1} - \mathbf{p_2}\|$$
$$r = \frac{r_1 \cdot r_2}{r_1 + r_2}$$
$$\mathbf{F} = (\kappa \cdot \delta - \gamma \cdot \sqrt{r \cdot \delta}) \cdot \frac{\mathbf{p_1} - \mathbf{p_2}}{\|\mathbf{p_1} - \mathbf{p_2}\|}$$

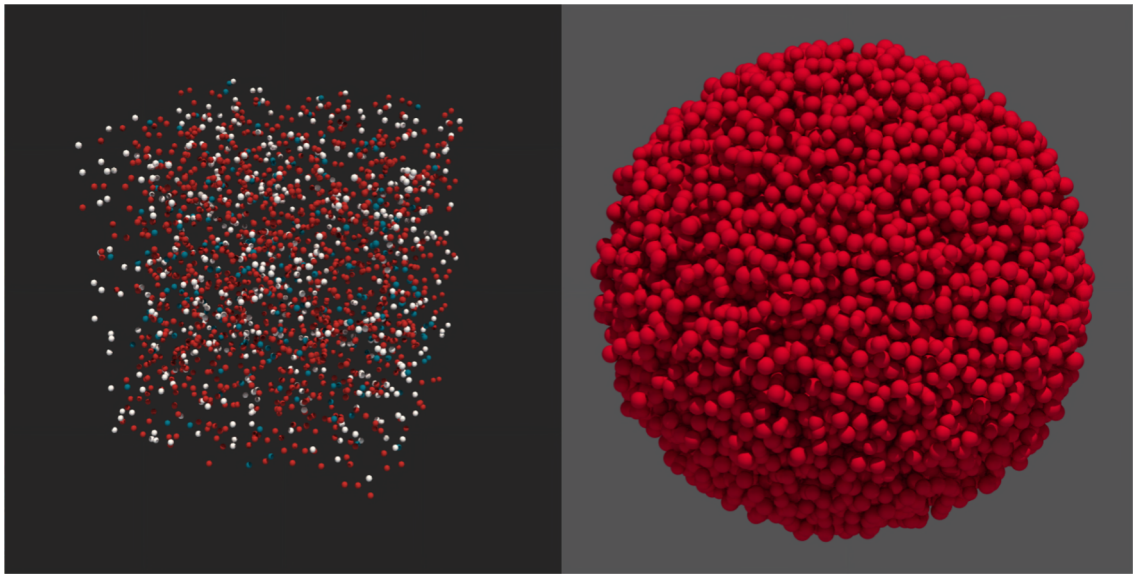
Collision force computation

Source: Hesam et al. 2021, https://doi.org/10.48550/arXiv.2105.00039

Performance Evaluation

Benchmark simulations





Benchmark simulations characteristics

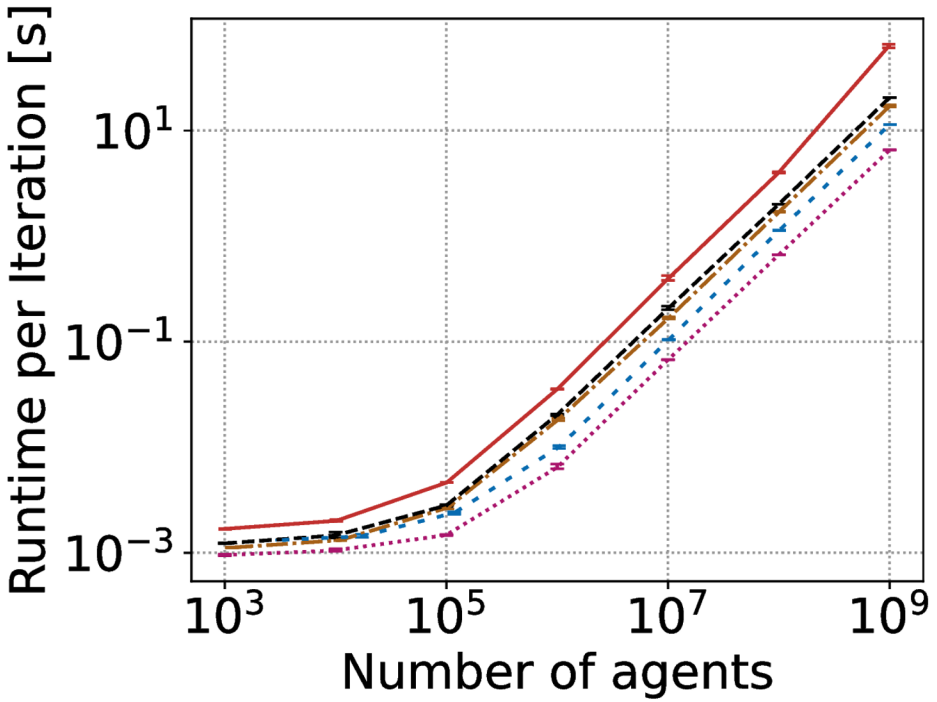
Characteristic	Cell proliferation	Cell clustering	Epidemiology use case	Neuroscience use case	Oncology use case
Create new agents during simulation	✗			✗	✗
Delete agents during simulation					✗
Agents modify neighbors				✗	
Load imbalance			✗	✗	
Agents move randomly			✗		✗
Simulation uses diffusion		✗		✗	
Simulation has static regions				✗	
Number of iterations	500	1000	1000	500	288
Number of agents (in millions)	12.6	2	10	9	10
Number of diffusion volumes	0	54m	0	65k	0

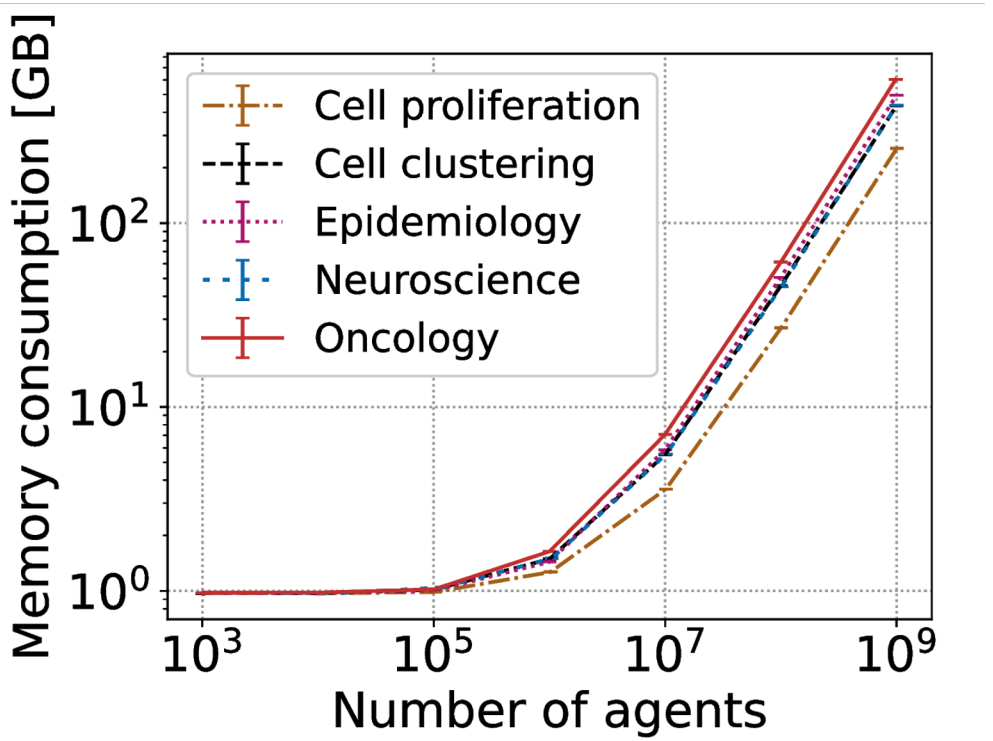
Benchmark hardware

TABLE II: Benchmark hardware

System	Main memory	CPU	OS
A	504 GB	Server with four Intel(R) Xeon(R) E7-8890 v3 CPUs @ 2.50GHz with a total of 72 physical cores, two threads per core and four NUMA nodes.	CentOS 7.9.2009
B	1008 GB		
C	62 GB	Server with two Intel(R) Xeon(R) E5-2683 v3 CPUs @ 2.00GHz with a total of 28 physical cores, two threads per core and two NUMA nodes.	CentOS Stream 8

Runtime and Memory Complexity

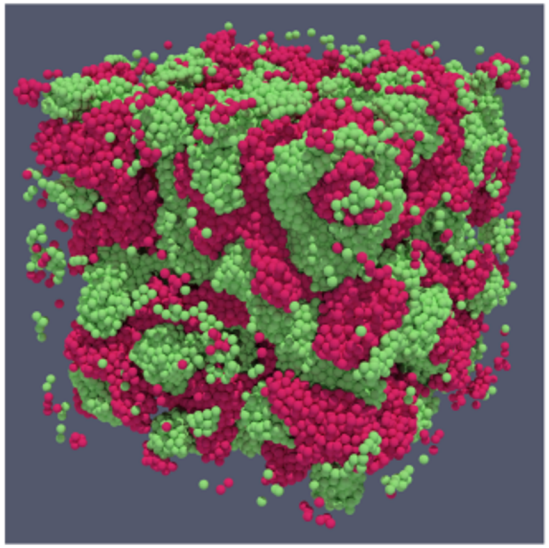


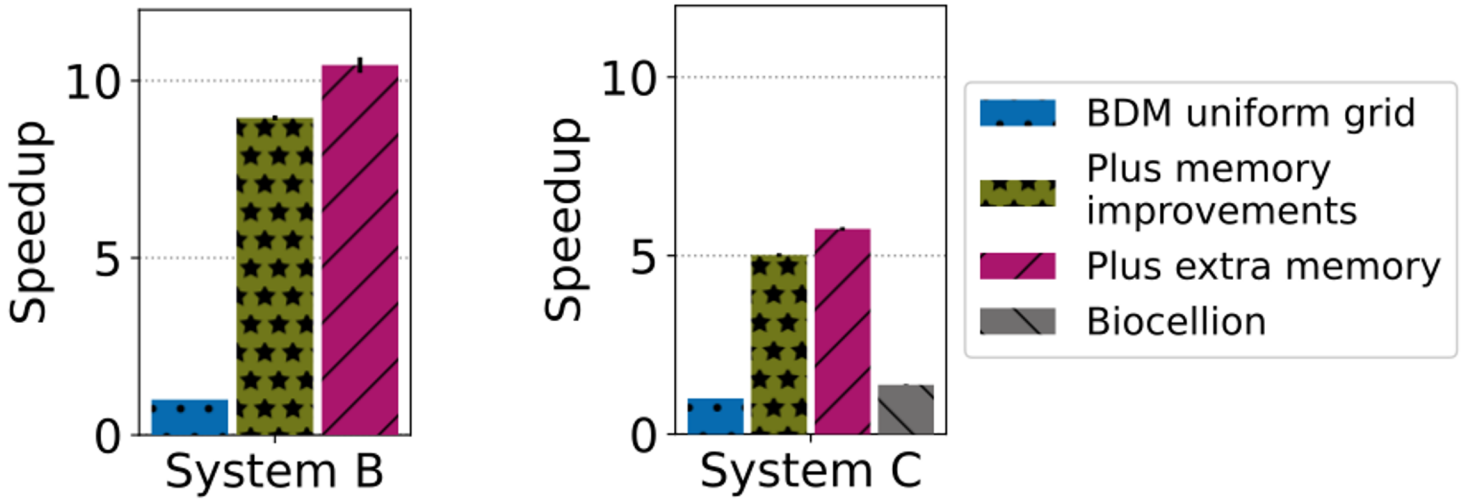


Comparison with Biocellion

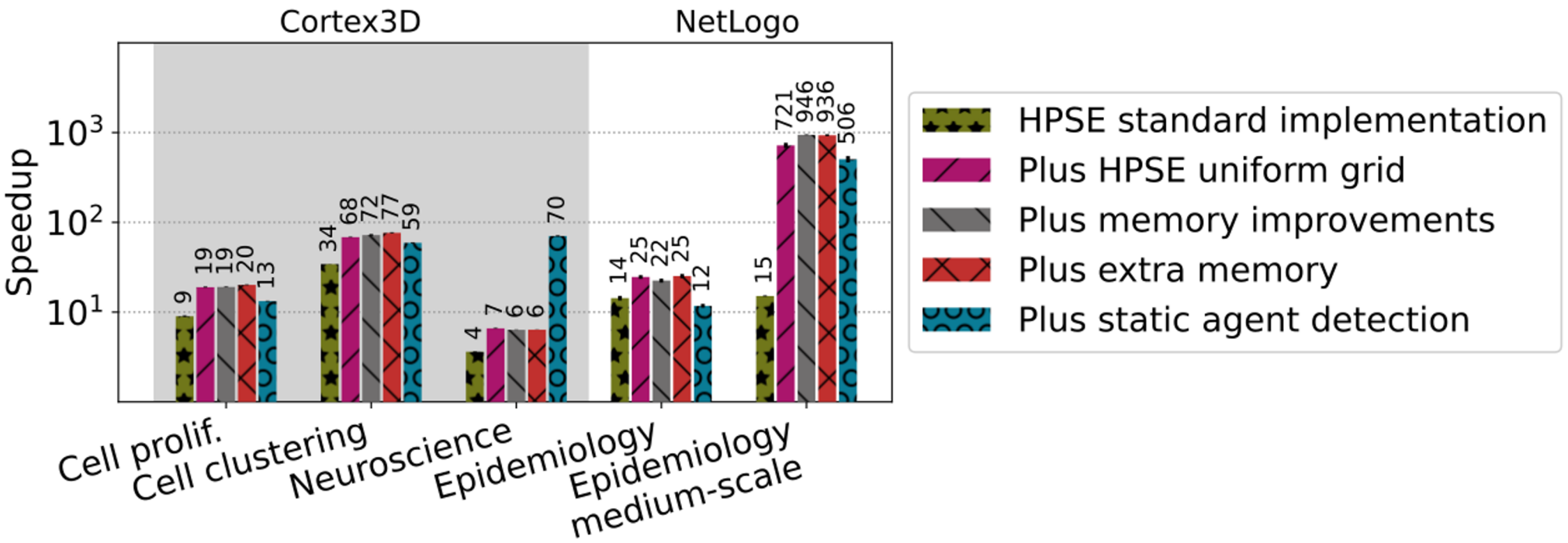
- Single-node 16 CPU cores; 13.4 million cells
→ BioDynaMo is **4.15x faster**

- Biocellion: 21 nodes, 672 CPU cores, 281 million cells
BioDynaMo: one node, 72 CPU cores
→ same runtime, but **9.3x fewer CPU cores** used

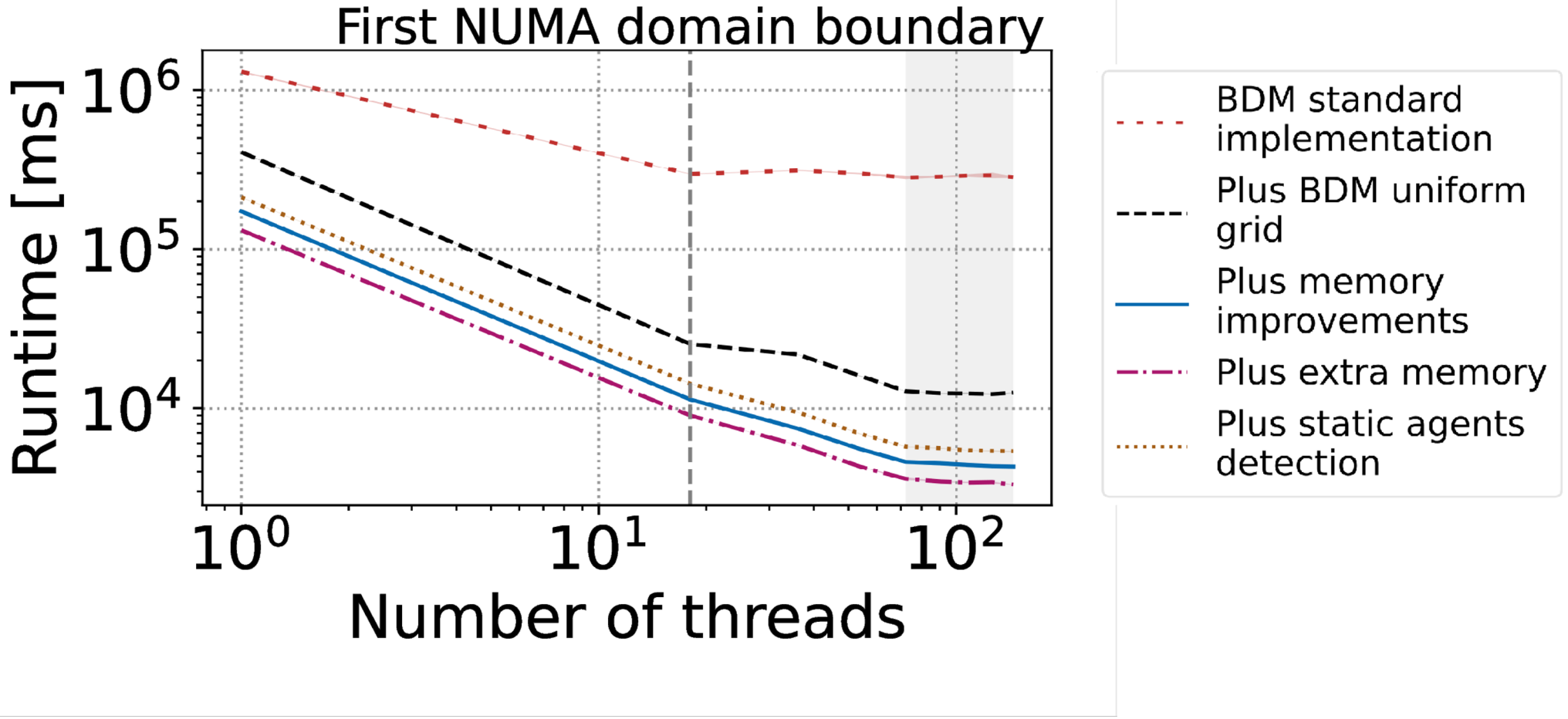




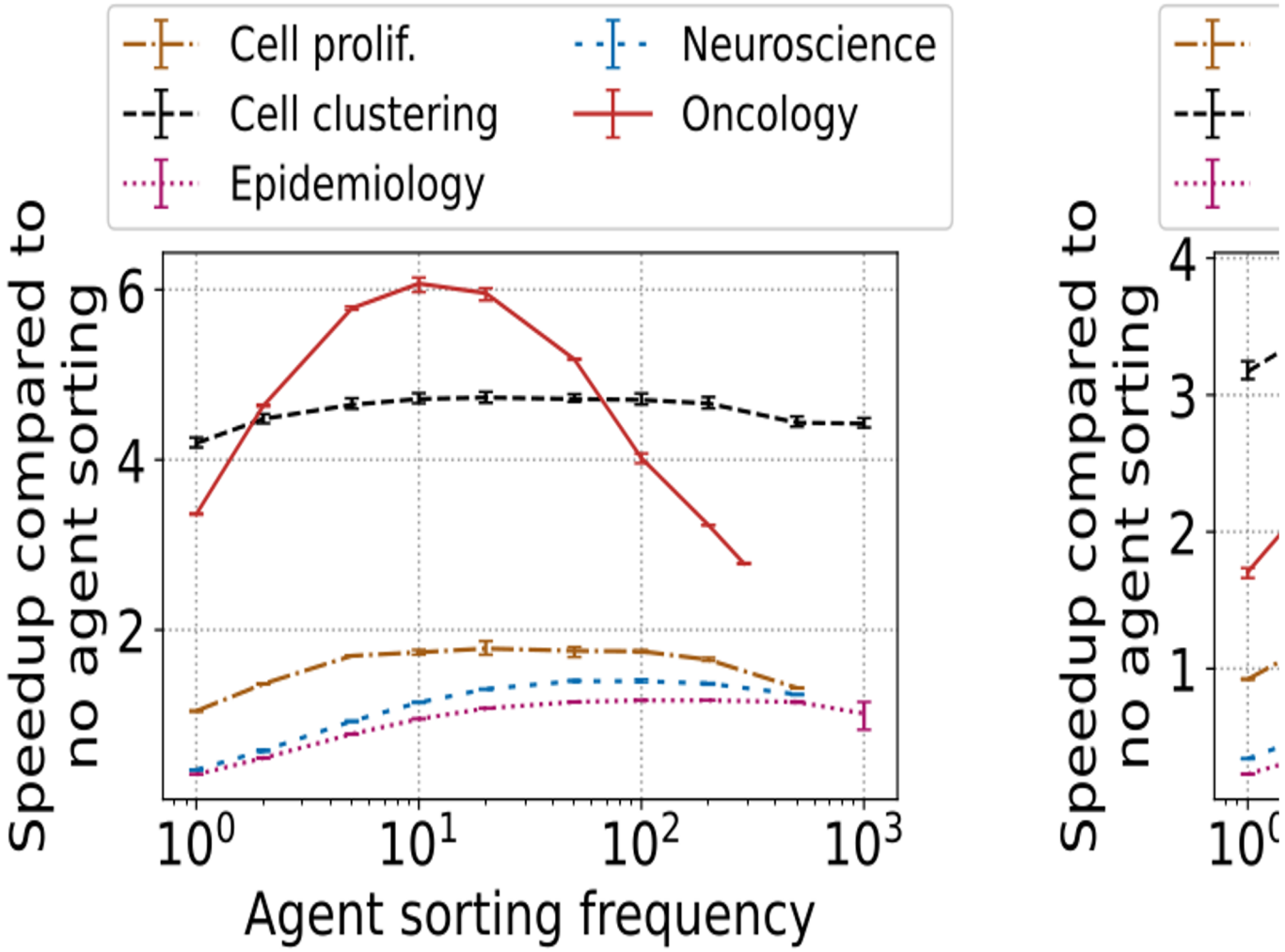
Comparison with Cortex3D and NetLogo



Scalability



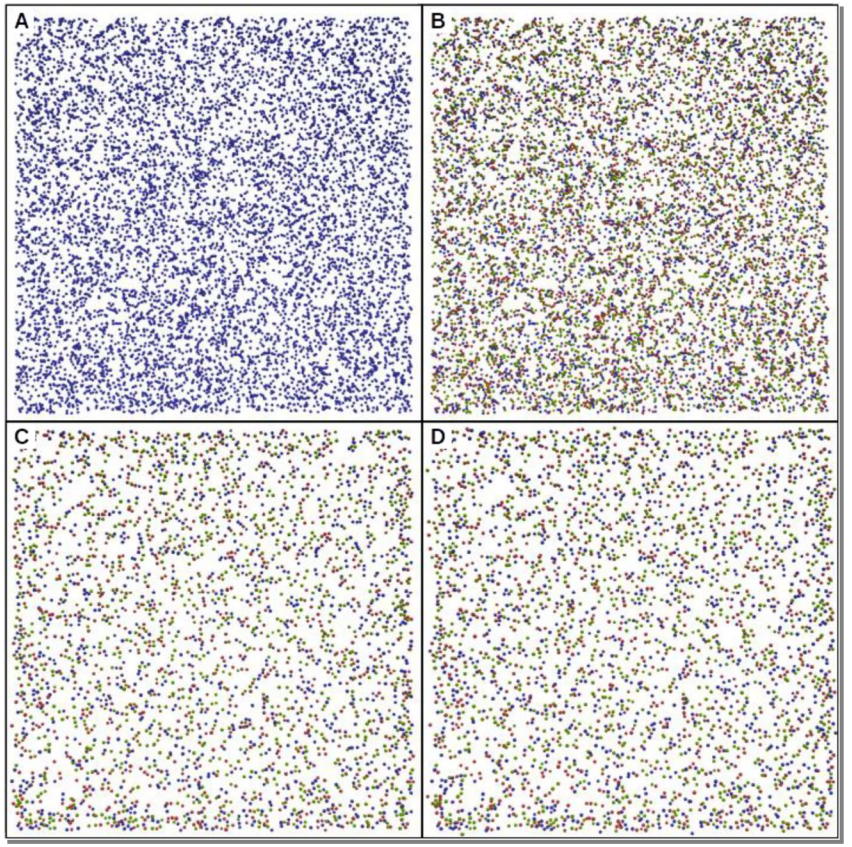
Agent Sorting and Balancing



Ongoing Use Cases

Retinal self-organization

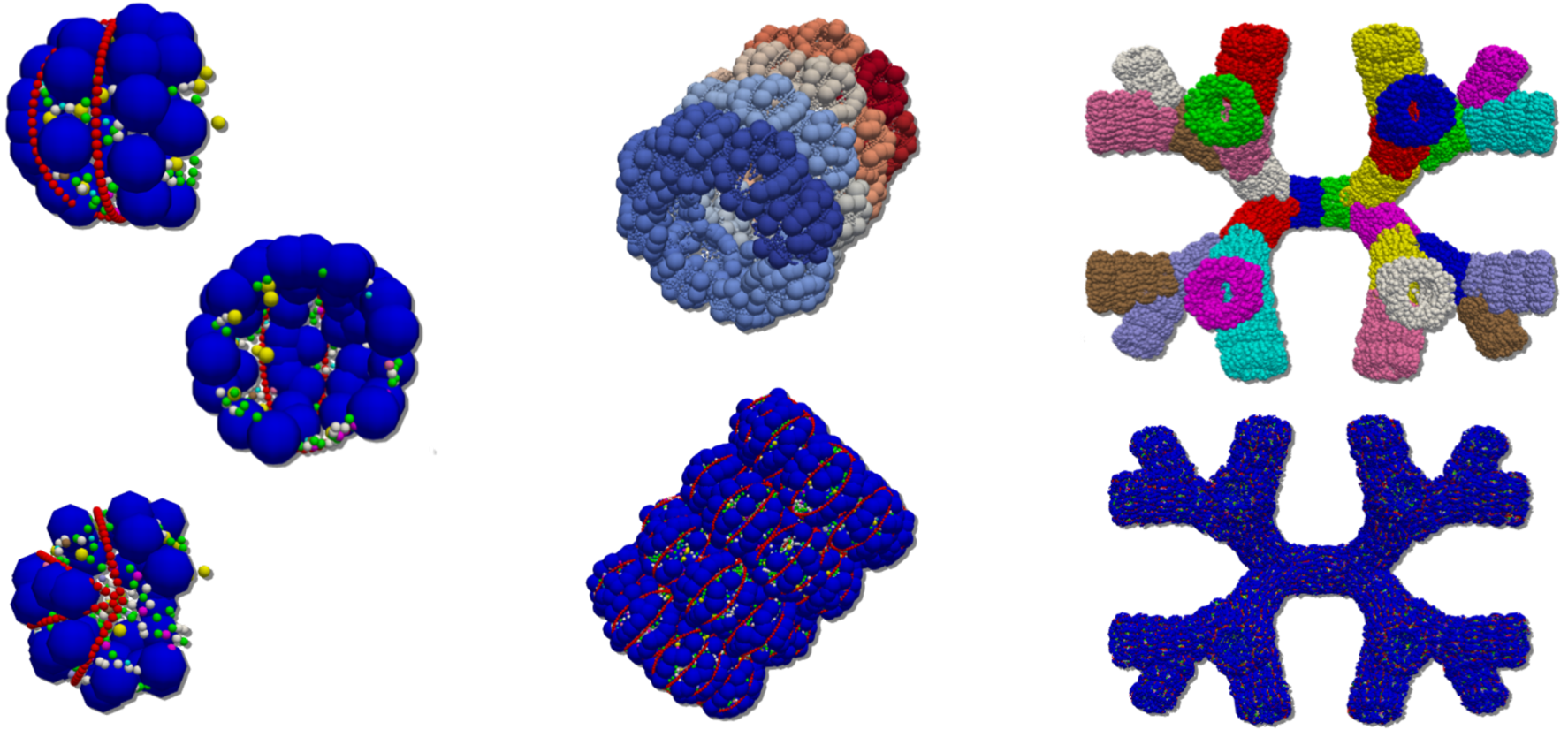
- Understand the mechanisms of cells self-organization during early development which is pivotal for their function.



Lead: Jean de Montigny, **Source**: https://www.biorxiv.org/content/10.1101/2021.10.22.465398v1

Radiation-induced lung injury simulation

- Simulate onset of radiation pneumonitis and/or lung fibrosis in normal tissue after exposition to thoracic irradiation.



Slide credit: Nicolò Cogno (GSI and TU Darmstadt)

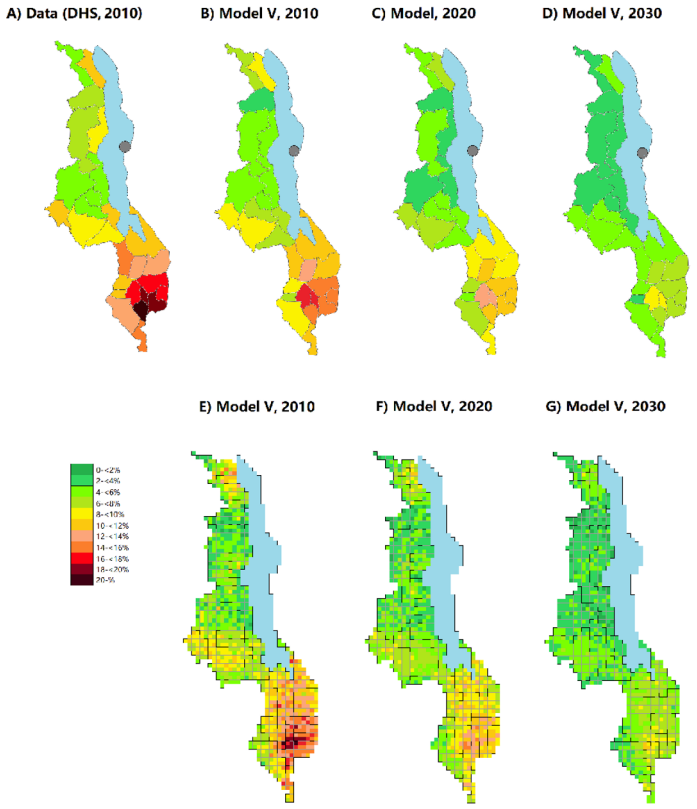
Spatial Spread of HIV in Malawi

- Collaboration with UniGE
- Original simulation written in R (Runtime: ~5.5h)
- Goal: speed up execution time
- Preliminary runtime with BioDynaMo: less than **2 minutes**
- Further work needed to make models equivalent

The spatial spread of HIV in Malawi: An individual-based mathematical model

Janne Estill, Wingston Ng'ambi, Liudmila Rozanova, Olivia Keiser

doi: https://doi.org/10.1101/2020.12.23.20248757



Lead at CERN: Tobias Duswald. **Partners at Uni Geneva**: Aziza Merzouki, Janne Estill. **Source**: https://www.medrxiv.org/content/10.1101/2020.12.23.20248757v3

Summary

- Agent-based simulation can be used to model many systems
- The presented optimizations improve the performance over state-of-the-art
  - Up to **three orders of magnitude speedup**
- These improvements allow BioDynaMo simulating **billions of agents** on a single server
- BioDyanaMo is currently being used in:
  - neurosience
  - oncology
  - epidemiology
  - cryobiology
  - socioeconomics
  - finance
  - ...

Thank you for your attention!

Lukas.Breitwieser@cern.ch

P&S Heterogeneous Systems

Accelerating Agent-Based Simulations with BioDynaMo

Lukas Breitwieser
ETH Zürich

Fall 2022

30 January 2023